Twitter Sentiment Analysis: Test Plan

Prepared by: Khalid El Essawi, Prabhav Arora, Saad Teeti
(5th December 2020)

TABLE OF CONTENTS

1.0 INTRODUCTION

2.0 OBJECTIVES AND TASKS
- 2.1 Objectives
- 2.2 Tasks

3.0 SCOPE

4.0 Testing Strategy

5.0 Test Schedule

6.0 Resources/Roles & Responsibilities

7.0 Schedules

1.0 INTRODUCTION

A brief summary of the product being tested. Outline all the functions at a high level.

2.0 OBJECTIVES AND TASKS

2.1 Objectives

Description of the objectives for the testing plan:

- Test code and ensure desired functionality from previously identified use cases is achieved
- Remove bugs and errors from code
- Ensure software is coded as efficiently as possible
- Identify any additional functionalities that are required for the software to run
- Ensure user experience of software is as smooth and user friendly as possible

2.2 Tasks

1. **Test Case Creation**: Identify use cases and then create test cases for each use case
2. **Testing Phase**: Test code with the previously identified use cases.
3. **Post-Testing:** Analyse the results of the testing phase and fix bugs and code accordingly
4. **Problem Reporting:** Reporting if any problem has been encountered during the process of testing.

List all tasks identified by this Test Plan, i.e., testing, post-testing, problem reporting, etc.

3.0 SCOPE

General

This section describes what is being tested, such as all the functions of a specific product, its existing interfaces, integration of all functions.

Our Key Functionalities

1. Customer enters a keyword of interest
2. Software connects to Twitter server to retrieve and preprocess latest tweets
3. Software does NLP Sentiment analysis on tweets
4. Software produces Sentiment Score and export reports as excel file for the table and text file for the foundings

Tactics:

List here how you will accomplish the items that you have listed in the "Scope" section.

The first key functionality is the customer being able to enter a keyword of interest.

- **2 test cases to test functionality**
- Test 1a will test keyword functionality by inputting spaces and empty strings. Expected Result: Error in program
- Test 1b will test keyword functionality by inputting symbols and strings with spaces such as “paris is a nice country”. Expected Result: Program retrieves tweets with these varied inputted tweets

The second key functionality is our software connecting to the Twitter server to retrieve and preprocess latest tweets:

- **4 test cases to test functionality**
- Test 2a will test to ensure that the maximum amount of available tweets are being extracted from the server. We will pass popular keywords (ex. “Messi”) and expect 100 tweets. We will then pass weird keywords (“hwljelqwejqwq”) and expect very few tweets. We will also go on twitter server to verify
- Test 2b. One of our sub-functionalities is allowing users to select the number of tweets to retrieve. To verify this functionality is operational our test will specify different amount of tweets and we will check how many tweets have been retrieved
- Test 2c will check all tweets retrieved from twitter server contain the keyword inputted. Expected Result: 100% of tweets retrieved contain user keyword.
- Test 2d will verify the recentness of tweets retrieved. Expected result: popular topic tweets should have timestamp of < 1hr. Old topics should have longer timestamp

The third key functionality is that our software does NLP Sentiment analysis on tweets.

- **1 test case to test functionality**
- Test 3a will test to verify valid NLP analysis is being conducted. Software predicted tweet sentiments will be compared with human labelled sentiment. Expected Result: >=75% accuracy of predictions

The fourth functionality is that our software produces Sentiment Score and export reports as excel file for the table and text file for the findings.

- **1 test case to test functionality**
- Test 4a will check whether sentiment score excel report and text file has been generated correctly

4.0 Testing: System and Integration Testing

Definition:

Our understanding of System and Integration Testing was based off of the requirements below:

- Correct acquisition of all data by the software
- Scaling and range of data as expected from software
- Correct output of data from software to hardware
- Data within specifications
- Interrupts processing
- Timing

Participants:

Khalid, Prabhav, Saad

Methodology:

Regarding the system and integration testing, the first thing we wanted to go ahead and test was how big of a training set we can use to train the model of the program. To do this, we kept factors such as the system used, and internet connection to be constant. More specifically, we wanted to see how much RAM building the training set would consume for different sizes. To test this, we used training sizes of 14k, 17k, 20k,. A **FAIL** in this case is if we are not able to input a training model of more than 17k due to not enough space on the RAM. Secondly, we tested out if our system correctly takes the keyword as input by the user. We tested this out by trying to type in strings of different lengths, characters such as ”Egypt”, “Egypt is”, “@##$%”, as well as an empty string “” and a string of spaces for e.g: “ “. The **PASS/FAIL** criteria for this test was all the keywords except the empty string and string full of spaces should move forward to the “connect to twitter servers and retrieve tweets” process, however the empty string and string full of spaces should display an error. The next thing we wanted to test was if there was a correct acquisition of all data by the software, being that if it is successfully able to retrieve 100 tweets based on the keyword successfully after the user enters a keyword. The way we set out to test this was by typing in three key words, the first being “Messi” as he is a very famous football player and we were sure that there existed 100 recent tweets of him, the second keyword was a less popular tweet that we suspect not to have 100 tweets retrieved even though we set the maximum number of tweets to 100 “knafa”, and the third keyword we input was just random letters of the alphabet that we were sure there were no recent tweets of, such as “dnsdniendeionwoenfnfowenfnfon”, unless somebody has typed that of course. For this test, the **PASS/FAIL** criteria is if the system was successfully able to retrieve the 100 latest tweets for the first 2 keywords and zero tweets for the last keyword from Twitter based on the keyword typed in by the user. Then we went ahead and tested the fetching of the tweets for a wide range of a number of tweets, in an attempt to see how our program would react to different scaling and range of data. We set the number of tweets to be fetched to 50, 100, 200. The **PASS/FAIL** criteria for this test was if the system was able to fetch the latest fetch at least 100 tweets and 50 tweets. Our next test was to check if the tweets that we are fetching are recent tweets. We did this by inputting the keyword (Barcelona Cadiz), to which we expect our program to extract tweets about the most recent football game which happened on the 5th of December between FC

Barcelona and Cadiz. The **PASS/FAIL** criteria for this test is if the tweets are around the time of the football game, to check this we look at the date the tweet was tweeted, a **PASS** being if the tweets are tweeted around the time of the football game and discuss the football game as well. And finally, we then set out to see if the keyword we are inputting correctly matches the tweets being retrieved, to do this we input a word, for e.g: “Palestine”, and **PASS** in this test would be if each tweet that is retrieved includes the world “Palestine”. We then moved on to the “sentiment analysis” part of the program, the first test, we set out to see how accurate our sentiment analysis NLP program, to do this, we ran “Ronaldo” as the keyword and we set the number of tweets to 10. We then perform a human cross validation of the results ourselves and we compare that result to the result of the program. A **PASS** in this case would be if our results are equal ±25%. The last and final functionality we wanted to test out was if our program successfully generates the report by exporting an excel file with all the tweets retrieved as well as a .txt file with the sentiment analysis score, if it does this successfully then it **PASSES** the test. This was tested by running the keyword “Dior” and running the program. We then try running the program again with the same keyword and check if it overwrites both files with the latest 100 tweets and sentiment analysis score, making sure these tweets are different, if it does this successfully then it gets a **PASS**.

5.0 TEST SCHEDULES

Test	Phase	Participants	Time
Size of Training set Test (0a)	Training the model	Saad	12:30 PM
Correct Input Test (empty string) (1a)	Keyword Input	Prabhav	12:45 PM
Correct Input Test (string with spaces) (1a)	Keyword Input	Prabhav	1:00 PM
Correct Input Test (symbols: !@#$%) (1b)	Keyword Input	Khalid	1:15 PM
Correct Input Test (different length of strings) (1b)	Keyword Input	Khalid	1:30 PM
Correct Number of Expected Tweets Retrieved based off keyword popularity (2a)	Tweet Retrieval	Khalid	1:45 PM
Correct Number of	Tweet Retrieval	Khalid	2:00 PM

Tweets Retrieved based on User Specification (for e.g user only specifies 20 tweets) (2b)			
Correct Tweets Based on Keyword Inputted (Matching) (2c)	Tweet Retrieval	Prabhav	2:15 PM
Correct Tweets based on Recentness (2d)	Tweet Retrieval	Khalid	2:30 PM
NLP Score Validity (Human Cross Check) (3a)	Sentiment Analysis	Prabhav	2:45 PM
Correct Report Generated (4a)	Sentiment Analysis & Twitter Retrieval	Saad	3:00 PM

6.0 RESOURCES/ROLES & RESPONSIBILITIES

Prabhav Arora :

1. Design Test Cases for Key Functionalities 1(Keyword Input) and 3(NLP Sentiment Analysis).
2. Conduct testing for functionality 1(Keyword Input)
3. Assist in drafting test results document

Khalid El Essawi :

1. Design test cases for key functionality 1(Keyword Input), 2(Retrieval of Tweets), 4(Correct Report Generated)
2. Conduct testing for functionality1(Keyword Input), 2(Retrieval of Tweets), 4(Correct Report Generated)
3. Assist in drafting test results document

Saad Teeti:

1. Assist in drafting test results document
2. Design test cases for key functionality: 0(Size of Training Set Test), 4(Correct Report Generated)
3. Code Implementation

Resources:

1. https://scikit-learn.org/stable/modules/generated/sklearn.naive_bayes.BernoulliNB.html
2. https://pythonprogramming.net/sklearn-scikit-learn-nltk-tutorial/

3. https://towardsdatascience.com/beginners-guide-for-data-cleaning-and-feature-extraction-in-nlp-756f311d8083
4. https://monkeylearn.com/blog/sentiment-analysis-of-twitter/
5. https://www.pluralsight.com/guides/building-a-twitter-sentiment-analysis-in-python
6. https://www.tweetbinder.com/blog/twitter-sentiment-analysis/
7. https://lionbridge.ai/articles/how-to-build-a-twitter-sentiment-analysis-system/
8. https://www.pluralsight.com/guides/building-a-twitter-sentiment-analysis-in-python
9. https://medium.com/better-programming/twitter-sentiment-analysis-15d8892c0082

7.0 SCHEDULES

Major Deliverables

Identify the deliverable documents. You can list the following documents:

- Test Plan
- Test Case
- Test Results